EFFICIENT MODIFICATION OF FAST UPDATED FP-TREES BASED ON PRE-LARGE CONCEPTS

Chun-Wei Lin¹, Tzung-Pei Hong^{2,3} and Wen-Hsiang Lu¹

¹Department of Computer Science and Information Engineering National Cheng Kung University Tainan, Taiwan { p7895122; whlu }@mail.ncku.edu.tw

²Department of Computer Science and Information Engineering National University of Kaohsiung Kaohsiung, Taiwan

³Department of Computer Science and Engineering National Sun Yat-sen University Kaohsiung, Taiwan tphong@nuk.edu.tw

Received August 2009; revised February 2010

ABSTRACT. In this paper, we propose a new FUFP-tree maintenance algorithm, based on the concept of pre-large itemsets, in order to efficiently handle record modification. Prelarge itemsets are defined by a lower support threshold and an upper support threshold. This helps to reduce rescans of the original database. The proposed approach can get good execution time for tree maintenance, especially when a small number of records are modified. Experimental results showed that the proposed Pre-FUFP modification algorithm performed well when handling updated records and generated nearly the same tree structure as the original FP tree algorithm.

Keywords: Data mining, FP-tree, FUFP-tree, Pre-FUFP algorithm, Pre-large itemsets, Record modification, Maintenance

1. Introduction. Years of data mining research has produced a variety of efficient techniques. Mining approaches find several different classes of knowledge including association rules, classification rules, clustering rules and sequential patterns [4, 16], among others. Finding association rules in transaction databases is the most common way in data mining [1, 2, 3, 6, 7]. The approaches can be divided into two categories: level-wise approaches and pattern-growth approaches. The apriori algorithm [1] was first proposed to mine association rules based on a level-wise processing method, which generated and tested candidate itemsets level-by-level. The FP-growth algorithm was then proposed to construct compressed tree structures and to mine rules without generating candidate item sets [8]. This was used to compress databases into tree structures that stored only large items. It was compact and complete with regard to finding all the frequent patterns. A recursive mining procedure called FP-growth was executed to derive frequent patterns from the FP-tree. They showed that the approach performed better than the apriori approach.

The FP-tree approach, however, needs to process all transactions in a batch way. In the past, Hong *et al.* thus proposed the Fast Updated FP-tree (FUFP-tree) structure to efficiently handle the newly inserted transactions in incremental mining [10] and also